

# Victorian 6502 User Group Newsletter

# KAOS

For People Who Have Got Smart

HARDWARE .. .. . DAVID ANEAR  
SOFTWARE .. .. . JEFF RAE  
FORTH .. .. . DAVID WILSON  
AMATEUR RADIO. ROD DRYSDALE VK3BYU  
EDUCATION .. .. . NOEL DOLLMAN  
LIBRARY. .. .. . RON KERRY  
TAPE LIBRARY .. .. . JOHN WHITEHEAD  
DISK LIBRARY WARREN SCHAECH (B.H.)  
NEWSLETTER. .. .. . IAN EYLES  
SYM .. .. . BRIAN CAMPBELL  
SECRETARY .. .. . ROSEMARY EYLES

OSI	SYM	KIM	AIM	APPLE	UK101	ORANGE
-----	-----	-----	-----	-------	-------	--------

Registered by Australia Post  
Publication No. VBG4212

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Vol.3 No.6

March 1983

If you have been unable to contact Jeff Rae it is because he has moved again and is living in temporary accomodation until his new house is built, so for the next three months his phone number will be 03 317 9481.

You are probably aware that since we started using the school for meetings some members have come along early each meeting to give the grade 5 and 6 children some experience in using computers. Unfortunately, we found that this wasn't very successful as some of the children were not able to come every month. This made it very difficult to plan any sort of program for them and so most of the time was spent playing games.

A new principal has taken over the school this year and he is interested in using computers in schools. After talking to him and one of the teachers who has enough knowledge of computers to be able to help the children, I decided to lend the school my spare Series II Board, then a few days later the club had the chance to buy a Series I Board quite cheaply, so now the school has two Superboards on extended loan, a standard Series II and a Series I with 48 X 24 screen format. It is not intended to teach the children to be computer experts but to give them a basic knowledge which will be useful when they get to secondary school. The principal would also like programs to assist those children who need extra help, especially with spelling and arithmetic. Any children who show particular interest will also be coming to one meeting each term, where hopefully they will be able to pick up some pointers from the experts.

Now for the commercial:

WANTED For Essendon Primary School.....Simple Educational Programs  
Do you have any, or can you write some for us?  
Please contact KAOS or Jeff Kerry

The next meeting will be at 2pm on Sunday 27th March at the Essendon Primary School on the corner of Raleigh and Nicholson Streets, Essendon.

The closing date for material for the next newsletter is the 8th April.

## INDEX

A/D Converter	11	Graphics - Microline 80	8
Comp-Dos 1.3	15	Machine Language pt 9	2
Dear Paul	3	Meeting Qld	6
Errata	12	Notes from Frank Halley	6+15
ETI Modem - Errata	7	Software Review	5
Fast Draw	4	Super Bexec*	13
For Sale	14	Superboard	4
Forth - Help Utility	16	The Meeting was KAOS	10

## THE BEGINNING MACHINE LANGUAGE PROGRAMMER.....Part 9

*by David Dodds*

Last month we laid out the broad plans for ARTIST and were considering the workings of the Make Pictures module.

Make Pictures you will recall started by calling a routine to input a valid key command. Let's assume that the command will be stored somewhere in memory by this GETKEY routine. The next phase of Make Pictures consists of a series of tests to determine which task has been requested. The test is done using a series of COMPARE instructions.

The 6502 allows comparisons between the contents of the accumulator or an index register and memory.

The mnemonics for these instructions are:

CMP    compare Accumulator and memory  
CPX    compare X    register and memory  
CPY    compare Y    register and memory

The comparison is made by subtracting the value stored in memory from the contents of the specified register. However the COMPARE instruction does not change the values contained in memory or in the register. Only the contents of the status registers change. The subtraction is actually carried out in the ALU (Arithmetic/Logic unit) and the status flags are set according to the results of the subtraction as follows:

N=1 if the final value was between \$80 and \$FF (ie minus) otherwise, N=0

Z=1 if the value subtracted was equal to the contents of the register  
otherwise, Z=0

C=1 if the value subtracted was less than or equal to the contents of the register otherwise, C=0

Note carefully these conditions. It may take some time to familiarise yourself with the operation of the carry flag as it is the opposite of what you might expect.

Consider the situation where A=\$F6 and the operation CMP #\$18 is performed. You might expect that \$F6 less \$18 would yield a negative result but that no carry (borrow) was required. The 6502 however performs the operation by taking the 2's complement of the value in memory (ie converting it to a negative value) and then adding the values.

The operation becomes:

                                  \$F6 =    11110110  
and the 2's complement of \$18 (00011000) =    11101000 (see BMLP part3)  
  = 1+ 11011110 so that the carry is set  
signifying that the value subtracted was smaller.

Typically a compare is followed by a branch of some description. Note that because 3 of the Flags are altered by these instructions that there are 6 valid branch tests (BPL, BMI, BCC, BCS, BEQ, BNE)

In Make Pictures each of the tests will be of a similar form

- 1 load a test value into one of the registers
- 2 compare it to KYDOWN (which is the place where the command is stored)
- 3 branch if not equal to the next test
- 4 if equal call the appropriate routine.

If this format were applied for example to the Cursor Up command it would look something like:

```
LDA #'U
CMP KYDOWN    (this could be either a page zero or
               absolute mode instruction depending
               on where KYDOWN is)
BNE (next test)
JSR CUR.UP
```

Using this format the entire top layer can be coded as:

68 ;MAIN LOOP OF	124	BNE W.BACK
70 ;	126	JSR WINRYT
72 ;PROGRAM	128 W.BACK	LDA #'B
74 ;	130	CMP KYDOWN
76 ;	132	BNE S.SAVE
78 JSR SYSDAT	134	JSR WINLFT
80 JSR SET.UP	136 S.SAVE	LDA #'S
82 INIZ JSR CLRS	138	CMP KYDOWN
84 JSR TOPLIN	140	BNE S.LOAD
86 MAKPIC JSR GETKEY	142	JSR SAVE
88 LDA #'U	144 S.LOAD	LDA #'L
90 C.UP CMP KYDOWN	146	CMP KYDOWN
92 BNE C.DOWN	148	BNE PUTCHR
94 JSR CUR.UP	150	JSR LOAD
96 C.DOWN LDA #'D	152 PUTCHR	LDA #'P
98 CMP KYDOWN	154	CMP KYDOWN
100 BNE CRIGHT	156	BNE ERASE
102 JSR CURDWN	158	JSR PAINT
104 CRIGHT LDA #'.	160 ERASE	LDA #RUBOUT
106 CMP KYDOWN	162	CMP KYDOWN
108 BNE LEFT.C	164	BNE FLASHC
110 JSR CURRYT	166	JSR CLEAR
112 LEFT.C LDA #',	168 FLASHC	JSR BLINK2
114 CMP KYDOWN	170 FINISH	LDA #ESC
116 BNE FRWRDW	172	CMP KYDOWN
118 JSR CURLFT	174	BNE MAKPIC
120 FRWRDW LDA #'F	176	JMP \$FE00
122 CMP KYDOWN		

Next month we will tackle the top line of the display; the graphic window.

DEAR PAUL,

This month I'm asking the questions. I have received a couple of questions that I can't answer, maybe someone out there can help.

The first question is from a member who has increased the RAM on his C1 and now finds that when an input statement occurs at the 8K point, the first character input is dropped: ie. INPUT A 12: PRINT A 2.

The second question is from a member who has interfaced a Tandy line printer 8 to his C1. There appears to be a timing problem as the computer is sending the first character to the printer before it is ready.

If you can help with these problems contact me through KAOS.

# Superboard

March, 1983.

Newsletter of the Ohio Superboard User Group, 146 York Street, Nundah, 4012.

---

## FASTDRAW by Fonzie Noes Hoo

This program enables complex graphics to be drawn on the screen - instantly !!! The routine is a M/C one which the first 8 lines of the program poke into that much used space in page 2. If you already have something there, the code is portable which means it can be put anywhere without change, including ROM. The remainder of the program generates some graphics for demonstration purposes.

The graphic is defined as a string, read in from data statements. The string can be up to 255 characters long. The format of the string is rigid and must be maintained. We will use the graphic defined in lines 250 and 260 to explain how to build up a string. The first, third, fifth and all odd numbered data are characters while the second, forth, sixth and all even numbered data are location information. Thus line 250 goes CHR\$(32), 1 space, CHR\$(210), 1 space, CHR\$(207), 30 spaces etc. As the Superboard screen is 32 characters wide, you have to wrap around to the next line of characters. Once you try a few for yourself on paper first, you will soon get the idea. At the end of line 250, the data could read 189,2,189 as CHR\$(32) is a space and is not printed. This only works, of course if there is already a space there on the screen, so I don't recommend it in practical applications.

To call the FASTDRAW routine, you define the position of the top left hand edge of the graphic on the screen with X and Y, being row and column, and the command then is C=USR(C)X,Y,A\$. A\$ is the name of the string graphic you wish to draw.

```
100 FORR=1 TO 30:PRINT:NEXT:POKE 11,161:POKE 12,2:REM FASTDRAW
120 DATA32,174,179,160,0,132,254,134,255,70,255,70,255,70,255,102
130 DATA254,102,254,102,254,32,2,180,138,24,101,254,133,254,165,255
140 DATA105,208,133,255,32,1,172,32,193,170,160,255,200,177,174,170
150 DATA150,172,192,2,208,246,136,136,132,95,177,173,145,254,230,273
160 DATA208,2,230,174,198,172,208,1,96,177,173,24,101,254,133,254
170 DATA144,2,230,255,230,173,208,2,230,174,198,172,208,220,96
180 FORR=673 TO 767:READ C:POKE R,C:NEXT
190 DATA161,1,161,1,161,1,161,29,161,32,161,1,161
200 DATA31,161,32,161,1,161,1,161,1,161
210 FORR= 0 TO 22:READ C:E$=E$ + CHR$(C):NEXT
220 DATA161,1,161,1,161,1,178,29,161,3,161,1,178,28,161,3,161
230 DATA1,161,28,161,3,161,1,175,28,161,1,161,1,161,1,175
240 FORR= 0 TO 32:READ C:D$=D$ + CHR$(C):NEXT
250 DATA32,1,210,1,207,30,32,1,209,1,208,30,189,1,32,1,189
260 DATA1,136,29,210,1,207,1,189,30,135,1,135
270 FORR= 0 TO 28:READ C:BX$=BX$ + CHR$(C):NEXT
280 X=15:Y=11:C=USR(C)X,Y,E$: REM DRAW E$ STARTING ROW 15, COL 11
290 Y=16:C=USR(C)X,Y,D$: REM DRAW D$ SAME ROW, COLUMN 16
300 X=22:Y=13:C=USR(C)X,Y,BX$: REM DRAW BX$
```

The beauty of this program is the speed with which you can draw multiple and complex graphics anywhere on the screen. Superb for fast games etc. X and Y can be incremented in loops where many characters are required. I can see many ready-made applications in Basic arcade-style games. To blank out a graphic, you can make up a BL\$ block, using CHR\$(32).

# — SUPERBOARD —

## SOFTWARE REVIEW - Victory Package.

You won't find many advertisements on the pages of SUPERBOARD or in its predecessor, the Newsletter. In this case however, a truly exceptional product is being offered, and it represents very good value for money. Many of the programs included have sold recently for \$14.95 each, and some of them are good enough to be worth it!

Like any collection of poems, pop music, or whatever, this product will undoubtedly contain programs that you already own, and many that you don't. This package offers something for everyone.

Most of the arcade games contain machine code subroutines to make smooth screen movements at realistic speeds.

BAR is a sort of mini-adventure game. Living patterns has machine code subroutines to avoid the long, boring waits that are generally associated with this type of simulation.

The Utilities are rather poor, much better ones are available to do the same things, but not at \$1 each!

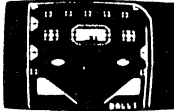
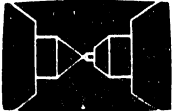


The statistics programs work just fine, but as no documentation is provided to explain their use, you need to know what to input, and what the answers mean.

Under the miscellaneous category is a truly mixed bag. Dehydration is a demonstration of how computers might be used in the medical profession. I did say "might" be used!

Blackjack Drill is compulsory for those computerists thinking of going to Wrest Point casino for a holiday.

The Victory package comes on two C60 cassettes, recorded both sides at 300 Baud.

Air mail to Australia would be A\$ 35. Highly recommended!

<h2>OSI</h2> <p>Stankiewicz &amp; Robinson, authors of MINOS, NIGHT RIDER, etc., proudly present to you:</p>		<h2>C1P</h2>	
<p><b>34 original PROGRAMS</b> on tape all for the unbelievably low price of <b>\$29.95!!</b> That's less than \$1 each!</p>			
 <p><b>PINBALL</b></p>	 <p><b>MINOS (MAZE)</b></p>	 <p><b>NIGHT RIDER</b></p>	 <p><b>RIDGE CRUISER</b></p>
<p><b>ARCADE TYPE</b></p>	<p><b>STRATEGY</b></p>	<p><b>UTILITIES</b></p>	<p><b>MISCELLANEOUS</b></p>
<p>NIGHT RIDER* COSMIC DEBRIS* MINOS* STREET SWEEPERS RIDGE CRUISER CAGE* PINBALL OSI GRAND* MINE FIELD WORM DEPTH CHARGE GOTCHA!</p>	<p>TAKE FOUR MIMIC MANCALA NEIGHBORS BAR LIFE FOR TWO*</p>	<p>TAPE VERIFIER LISTING LINE RE. # VERSATILE LINE RE. # LINE LOCATOR</p>	<p>MESSAGE ENCODER TYPING TUTOR PHONE NUMBER DEHYDRATION BLACK JACK DRILL</p>
	<p><b>KALEIDOSCOPIC</b></p>	<p><b>STATISTICS</b></p>	
	<p>LIVING PATTERNS KALEIDOSCOPE DRAW ME</p>	<p>CHI SQUARE FUNCTION PLOTTER BETTER RND. # GEN. PROBABILITY #1</p>	<p>(*Previously sold by AARDVARK™)</p>
<p>All programs will run on 8k C1P. Many are compatible for C2/4 and run in 4k.</p>		<p>Please add \$1.50 postage &amp; handling PA resident please add 6% sales tax Charge customers include # and expiration date</p>	
<p>(215) 296-3787</p>		<p><b>VICTORY</b> SOFTWARE</p>	
		<p><b>VICTORY SOFTWARE CORP.</b> 7 Valley Brook Road Pao11, PA 19301 USA</p>	

Queensland User Group Meeting. 20.2.83

Attendance 20

Computers 6

Started 12 noon

Finish 5pm

Dave Prince had a bare Ozi Rabble board which was admired by all. Dave planned to build it up slowly as finance permitted.

Robin Wells and Paul Brodie were attempting to get Robin's latest project, Premier's Screen Enhancement Kit, going to his satisfaction. This works with the Cegmon screen window to give virtually any format, 24 x 24, 32 x 48, 16 x 64 and 32 x 64 all being demonstrated. The board was a small, neat double sided job with plated thru holes.

Robin announced that Paul's modem was crook, which explained why it refused to co-operate at the last meeting. Next meeting for sure.

Bill Simpson was having problems with one of the school's first ClPs, bought in mid 1979. The machine was almost stock standard, and still had only 4k ram. Bill explained that a lot of memory was not necessary for students to learn programming, and the 4k helped discourage games. The main requirement was a simple programming language with no unnecessary complications like integer variables etc.

Brendan Vowles had his disk system working perfectly, and demonstrated the latest version of OS65D which had some really superb features.

Bernie Wills and John Burnham had constructed a home grown 32 x 64 system, using a 6545 video controller. It worked well, and only uses about half the number of chips on a Tasan board. Main problem was initialisation, and a special routine had been programmed into Bernie's monitor rom.

Bernie has no plans to market the system, but put together some notes and drawings for me. If anyone is interested, send OSUG a SAE size 9 x 4.

The project is inexpensive and offers 24 x 24, 32 x 32 or 32 x 64 and a 50 Hz flicker free display. It is not for the beginner however.

Harry Moores provided the main interest in the meeting. All conversation ceased as Harry's SC-01 voice synthesiser interface boasted that it was a fantastic machine, counted to ten, and made several other comments before launching into a recital of a bawdy verse of "Mary had a little bear". Harry had built it up using a PIA and Dick Smith's kit. Paul Brodie is keen to build one. Harry was the focus of attention for some time.

Bob Best and son Chris dropped in some computer boards and fans, which were well received by the troops. There was only a couple left after the meeting.

Next meeting will be in April some time, after I have recovered from my holiday.

Wanted: "BITS & PIECES". We are holding a Computer/ Electronics garage sale on April 16/17 and are looking for suitable items. Any members wishing to turn their bits and pieces into cash should contact Carl or Lauren Nielsen on  
COMPUTER CLINIC.

WANTED: Person familiar with machine code to write short diagnostic routines for the Ohio Computer. Please contact Carl Nielsen on or write to

---

NOTES FROM FRANK HALLEY

There is a bug in COMP-DOS 1.2 in the code for "BS nn". Under some conditions, the system will put the disk buffer at the beginning of the BASIC source code storage area (at \$327E) where the original OSI system put it. This overwrites your program and ...CRASH! The solution is simple. The offset in one branch instruction is wrong (at least in my version - it may have since been fixed). Location \$5D4F/\$7D4F/\$9D4F (24/32/40K versions) contains \$06 which should be \$04.

## ERRATA FOR THE ETI 644 MODEM

Our thanks to Neil Murray for this errata for the ETI 644 modem. This information was taken from the MICOM bulletin board with the kind permission of MICOM's president Geoff Halprin.

29/11/82

Rang ETI on errata for modem. There is plenty. Some in Dec. and some Jan. Highlights:- R10, R21, R26, R9, R18, R28 are 10K - they should be 22K to correct the bandwidth. Junction of C31 & R76 should go to the cathode of D14. Cut near IC20 & bridge. C18 (100N) shown to pin 3 on IC12, should go to pin 2 IC12. On IC4 reduce C5 from 1N5 to 680P. New filter diode layout in Dec. ETI. There is an R,C net across RL4, value I don't know yet. Holds relay for dialing. On setup using 600BD osc. only works on high band. Increase C10 to 22N to run at 300BD low. S0 link(LHS) undo bottom end & connect +6V. Guess there is more, but I don't know what. Yuk! Roger Adair.

30/01/83

It has been found that the ETI modem has some serious design flaws that cause erratic operation on several of its modes. For those who have experienced such problems here is a list of changes that have been tried on a number of modems and found to yield a large improvement in performance. FIX 1 - Change R9, R10, R18, R21, R26, R28 from 10K to 18K. This sets the data and reference channel low-pass filters to the correct frequency. (As published they are out by a factor of 2.)

FIX 2 - Change the high-pass filter diodes as follows:-

FILTER MATRIX:	COLUMN	0	2	3	1	
	Row A:	D	N	N	N	(D=diode)
	Row B:	D	D	N	N	(N=no diode)
	Row C:	D	N	N	D	(As viewed on board)
	Row D:	N	D	N	D	

These settings give a better trade off between amplitude response and group delay distortion.

FIX 3 - The 75 baud receive mode does not work on most modems due to a DC offset problem in the data reference channels. It has been found that this problem can be cured by switching 10nF capacitors across C5 and C9 when in this mode. A toggle switch or relay can be used for this purpose. Wiring should be kept as short as possible. It may be necessary to slightly adjust the value of one of the capacitors in some cases.

I would be interested to hear from anyone attempting these modifications. I am documenting them to send to ETI. My phone no. is 02 94 6520 Bruce Orr.

03/02/83

Assembled kit, Modification made

- 1 - Modified PCB so junction C31/R76 goes to IC20 pin 6 as on original circuit.
- 2 - R41 reduced to about 400ohms to give -18dbm to line.
- 3 - R2 increased to 390K to allow Adj. of O/P symmetry. Symmetry adjusted using audio oscillator.
- 4 - Filter diode CH3/F removed. All others as per circuit. Unit has been used in 300 BD originate mode only. Quality of lines I use are poor!

The only incorrect characters received were on 2/2/83, 4 characters in 2000 received were incorrect. No other incorrect characters have been received during connection to 'CBBS'. Geoff Trone.

## A SIMPLE GRAPHICS PACKAGE

*by Rodney Eisfelder*

Computer Pictures and graphs are generally made up of lines (known in the trade as vectors.) However some output devices such as video screens and printers are only capable of turning on dots (called pixels) arranged in a rectangular array of columns and rows (known as raster lines on a video screen.) One of the most basic tasks of a graphics driver for such a device is to convert vectors to the appropriate raster dots. This task is built in on the Apple-II and BBC micro computers and is known as vector to raster conversion. The subroutines listed below perform this task for OS65D (and COMP-DOS) driving a Microline-80 printer whose graphics enable each character position to be treated as an array of six pixels (two across and three down.)

The subroutine package is divided into four parts:

The initialization section is host-dependant. On entry the variables X and Y define the maximum size of the picture in pixels. In this implementation an integer array is used to store the picture in memory. The lower twelve bits of each integer represent two graphics characters on the Microline-80.

The Move/Draw Section is both host and device independant and converts 'Draw' commands into a series of 'Set Pixel' Commands. The algorithm used is similar to the 'Simple DDA' described on page 24 of 'Principles of Interactive Computer Graphics' (Second Edition) by Newman and Sproull.

The Set Pixel Section is (more or less) device independant and depends only on the method chosen to represent data on the host. In the current implementation it sets a bit in the array. For completeness, a Clear Pixel Routine has also been provided, but it is not currently used.

The last section is the Output Section. This is the only section that needs to be device dependant, although usually the method used to store the picture on the host will be designed around the characteristics of the device, requiring changes to the Initialization Section and the Set Pixel Section as well. In the current implementation it is possible to draw the complete picture or a rectangular window.

My aim in presenting this simple package of Subroutines is to enable you to write graphics programs which can be transported not only from one computer to another, but also from one output device to another. Some devices such as video screens may not require an output routine, but they can all share a common graphics interface.

This is not the first graphics package and certainly not the best ever written, however given the limitations of Challenger Basic, I think it is a good start for those trying to draw graphs.

```
10 X=90:Y=72:GOSUB30000                                'AUSTRALIA' PROGRAM
20 READX,Y:GOSUB500:GOSUB30500
30 READX,Y:IFX<0THEN100
35 GOSUB500
40 GOSUB31000:GOTO30
100 IFX=-2THEN20
110 GOSUB30100:END
200 DATA9,39,12,40,14,42,14,44,18,46,19,48,20,49,22,49,24,48
210 DATA25,46,27,44,28,43,29,39,28,38,27,36,25,36,26,35,25,35
220 DATA24,34,26,32,24,30,24,26,25,22,26,19,27,16,27,14,26,13,25,13
230 DATA24,12,23,10,22,9,20,9,19,10,18,13,17,15,15,17,14,19,13,21
240 DATA13,25,11,26,10,28,10,34,13,32,14,36,14,37,12,38,9,39
250 DATA-2,-2,30,40,30,43,31,43,32,42,30,40
500 S=32-X:X=Y-5:Y=S
505 X=X*2:Y=Y*3
510 RETURN
999 DATA-1,-1
```

```

5 SH=.06:REM set Resolution of Curve
6 INPUT"PETAL FACTOR";P
10 X=50:Y=50:GOSUB30000
20 DEFFNX(A)=R*COS(TH)*25+25
30 DEFFNY(A)=R*SIN(TH)*25+25
40 DEFFNR(T)=SIN(T*P)
50 LM=3.14159 : IF P=2*INT(P/2) THEN LM=LM*2
60 TH=0:GOSUB200:GOSUB30500:REM Move to Start (with "Pen" Up)
70 FOR TH=SH TO LM STEP SH
80 GOSUB200:GOSUB31000
90 NEXT TH
95 TH=LM:GOSUB200:GOSUB30500:REM Draw back to start
100 GOSUB30100:END:Print Finished Picture
190 REM This subroutine computes X and Y on the curve being drawn.
200 R=FNR(TH):X=INT(FNX(TH)):Y=INT(FNY(TH)):RETURN

```

GOSUB 30000 - Initialization Section. Must be called first and called once only.  
 On Input X and Y define the largest possible values for calls to the graphic  
 package. Zero is implicitly the lowest. 06 and 07 are saved for future reference  
 GOSUB 30100 - Output Section. Prints the entire array. No Parameters.  
 GOSUB 30200 - Alternative Entry Point for Output Section. X1 and Y1 define the  
 lower limits of the area to be printed. X2 AND Y2 define the upper limits  
 GOSUB 30500 - Move Command X and Y are the co-ordinates of the point to be moved  
 to Move means "Go with the Pen Up" i.e.the next vector starts at this point  
 GOSUB 31000 - Draw Command X and Y are the co-ordinates of the point to be drawn  
 to (with the pen down.) They are also the start point for the next vector.  
 GOSUB 32000 - Set Pixel Command X and Y are the Pixel to be set.  
 GOSUB 32010 - Set Pixel - Internal Entry Point 08 and 09 are the coordinates of  
 the Pixel to be set.  
 GOSUB 32100 - Clear Pixel Command X and Y are the co-ordinates of the Pixel to be  
 cleared.

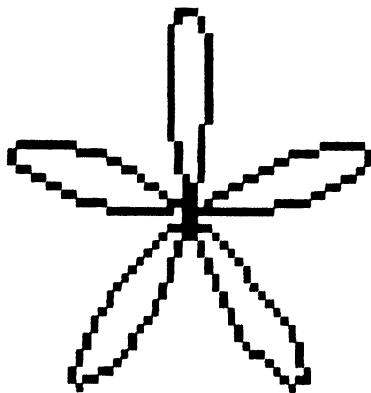
```

29999 STOP
30000 DIMO1%(X/4,Y/3):O6=X:O7=Y:RETURN
30100 O4=O6:O5=O7:O2=0:O3=0:GOTO30210
30200 O4=X2:O5=Y2:O2=X1:O3=Y1
30210 FOR O9=INT(O5/3) TO INT(O3/3) STEP -1
30215 PRINT#1,TAB(Z);
30220 FOR O8=INT(O2/4) TO INT(O4/4)
30230 PRINT#1,CHR$( (O1%(O8,O9)AND63)+128 );CHR$(INT(O1%(O8,O9)/64)+128);
30240 NEXT O8:PRINT#1:NEXT O9
30250 RETURN
30500 O2=X:O3=Y:RETURN
31000 IFABS(Y-O3)<ABS(X-O2)THEN31500
31010 O0=(X-O2)/ABS(Y-O3)
31020 O8=O2+.5-O0
31030 FOR O9=O3 TO Y STEP SGN(Y-O3)
31040 O8=O8+O0:GOSUB32010:NEXTO9:GOTO31600
31500 O0=(Y-O3)/ABS(X-O2)
31510 O9=O3+.5-O0
31520 FOR O8=O2 TO X STEP SGN(X-O2)
31530 O9=O9+O0:GOSUB32010:NEXTO8
31600 O2=X:O3=Y:RETURN
32000 O8=X:O9=Y
32010 O4=O1%(O8/4,O9/3)
32020 O4=O4ORINT(2*((O8AND1)+(O8AND2)*3+4-(INT(O9)-INT(O9/3)*3)*2)+.5)
32030 O1%(O8/4,O9/3)=O4
32050 RETURN
32100 O8=X:O9=Y
32110 O4=O1%(O8/4,O9/3)
32120 O5=INT(2*((O8AND1)+(O8AND2)*3+4-(INT(O9)-INT(O9/3)*3)*2)+.5)
32130 O4=O4ANDNOTO5
32140 O1%(O8/4,O9/3)=O4
32150 RETURN

```

GRAPHICS PACKAGE for the  
 MICRO-LINE 80

*Next page please.*



These illustrations were printed on a Microline 80 using Rodney Eisfelder's graphics package. They have been photo-reduced to half size.

---

THE MEETING WAS KAOS  
*by King Corky*

Well, I'm back again after a short rest in a hospital bed, and expensive they may be, but the service is great and the skill and expertise of the surgeons would be cheap at twice the price. You never really appreciate people like doctors, nurses, surgeons and especially nursing aides, until you see some of the things they have to cope with in a hospital. Anyway, back to more mundane things, it's nice to know that KAOS managed to get along without my illustrious presence. Thanks to Michael Lemaire for filling in.

Stuart Thomas has done it again with another ingenious but simple mod for our lowly!! OSI PC's, an A/D interface for storing in RAM, audio/ speech/ music or whatever. The hardware consists of just two (2) wires and the M/Code routine is about 200 bytes and will allow almost perfect? digitising of humun speech that can be replayed at any speed, forward, backwards or inside out. He has also written a game of POOL and would like to sell copies (he needs the money to finance more mods like the A/D). We also saw a floppy disk controller board for the Peach that costs around \$190.00. Not bad for a board that will fit in your shirt pocket eh ??

COMP-DOS 1.2 ROM (\$E800) is now available for systems running 5.25" (133.35mm) disks. Has DABUG, printer driver and runs with OS65D 3.2 or 3.3. A 40 to 48 pin bus adapter is also available at \$8 from David Anear, (bow, scrape & I kiss your hand). David, (praise, cheer & I kiss your hand again), demonstrated his Hi-Res system now in a stand alone case no bigger than a kleenex box.

KAOS has donated a Superboard to the school for allowing us the use of their rooms at no charge and now we need some educational, but entertaining, programs to go with it. The library has some but we need more. All you software genii out there, hop to it.

The RABBLE boys from Shepparton are doing wonders with their updated single-board system. At around \$400, (+-\$100 or so), it will rival any computer, single board or system, on the market, price notwithstanding.

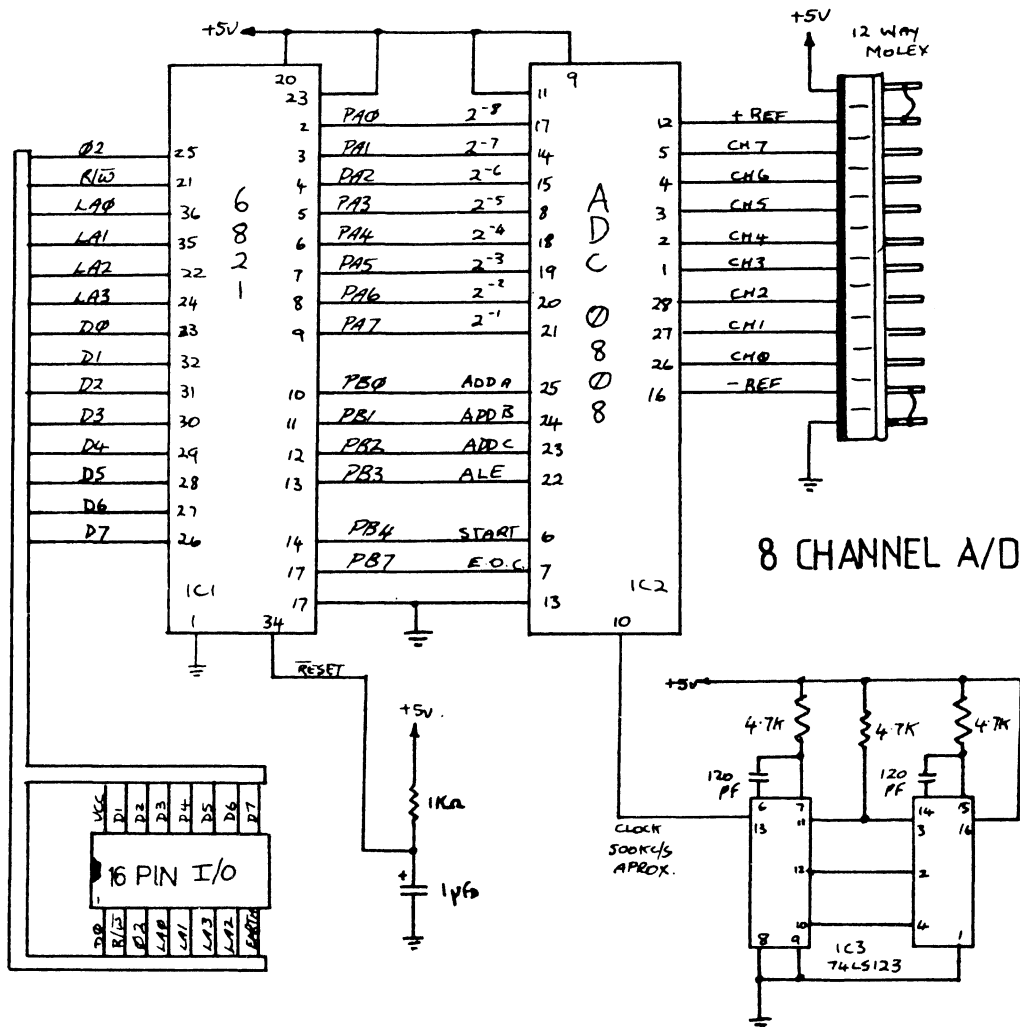
No potential 'SAGA' has emerged since GT-BUG, though I have yet to see George's assembler, so the bedtime story section will have to rest till one rears it's ugly head! I need something juicy to add more color, (colour), to the column. Somebody out there start a project big enough to capture the interest and imagination, but complex enough to make it impossible to complete, that will keep me going for months. Bye.

# 8 CHANNEL A/D CONVERTER

by David Anear

Faced with the problem of needing an 8 bit A/D converter for a special application (ie. a digitizer or mouse), I designed the following circuit for the 16 pin I/O Buss which was discussed in Nov. KAOS. The circuit uses the National A/D converter which has a built in 8 channel multiplexer.

The converter professes to be micro-processor compatible, but its slow access time on conversion made it easier to drive it from a PIA than to mess around with latches etc.



The A/D works like this:-

Connect the voltage you wish to measure to the channel you wish to use. (The voltage must be between 0V and +5V.)

Next put the BINARY code for the channel number you are using on ADD A, ADD B, ADD C.

Toggle the ALE line to load that channel number into the multiplexer.

Toggle the START line to start the A/D converter and when the E.O.C. line goes from low to high, the conversion is finished.

You then read the value from the A/D for the channel.

I have included a short BASIC program to read each channel and print the results on the screen.

To change the range of channels looked at, change the value of X, therefore, to look at channels 0 and 1 change line 110 to:-

```
110 FORX=0TO1:GOSUB200:GOSUB300
```

Next page please.

By connecting a pair of pots across +5V and earth, with the wiper to the desired channels, an analogue paddle or joystick can be easily made.

This description is very brief, so for more information contact me or get the data sheets on the ADC 0808.

16 pin I/O adapter cards for the Tasker Buss system can be obtained from David Tasker and 40 pin to 16 pin I/O boards from myself.

By removing the PIA and changing the I/O, the chip can easily be added to the proto area of the Rabble boards.

For those people who are going to ask why I did not divide 02 to get the clock for the A/D, the answer is that my system runs NMHz, therefore the clock switches from 1MHz to 2MHz causing problems for the A/D. Conversion time is 100uS approximately.

```
10 PO=50956:CO=PO+2
20 REM INITIALISE *****
30 POKEPO+1,0:POKEPO+3,0:POKEPO,0:POKECO,127
40 POKEPO+1,4:POKEPO+3,4:POKECO,0
100 REM MAIN ROUTINE *****
110 FORX=0TO7:GOSUB200:GOSUB300
120 PRINT"..";AN;:NEXTX:PRINT:GOTO100
200 REM CHANNEL SELECT *****
210 POKECO,X:POKECO,X+8:POKECO,X
220 RETURN
300 REM START AND READ RESULT *****
310 POKECO,X+16:POKECO,X:IFPEEK(CO)<128THEN310
320 AN=PEEK(PO):RETURN
```

---

#### RABBLE SOUND GENERATOR - ERRATA *by Paul Dodd*

There were two mistakes in my article (page 13 of last issue). First one was a typing error (either Rosemary's or mine - I can't remember if I had a hangover that day or not - come to think about it, I don't remember the last day I didn't have a hangover but that's another story - or several other stories).

Line 30 should read:

```
30 POKE PIA,255:POKE PIA+2,255
```

If you have a cassette machine, line 50 should read:

```
50 PSG=240:REG=PSG+1:INF=REG+1
```

(ie. delete the 12153)

#### EMBEDDING MACHINE CODE IN BASIC PROGRAMS--ERRATA *by M. Lemaire*

There was an omission in last month's article; another null is needed when manually poking the new start address for the program. The amended routine is shown here as a program for clarity.

```
10 A=PEEK(120)+PEEK(121)*256:REM Initial BASIC start addr.
20 INPUT"HOW MANY BYTES DO YOU WANT BELOW THE PROGRAM?";N
30 A=A+N
40 REM Split A into high & low byte values..
50 HB=INT(A/256):LB=A-HB*256
60 REM Poke in new values; set up 2 nulls..
70 POKE120,HB:POKE121,HB:POKEA-1,0:NEW
```

SUPER BEXEC\* FOR COMP-DOS 1.2  
*by Robert Bretterecker*

I can't take all the credit for what really amounts to a combination of a few good ideas.

The program listed below, while containing all the goody goodies of George's DOS, will boot up a machine with 24, 32 or 40 K's of memory without changing anything.  
It will print out a numbered "Menu" program.

To run a particular program on your disk, all you do is enter the appropriate number and carriage return. If carriage return is hit without first entering number, you will be brought into the immediate Basic mode.  
0 carriage return = "NEW"

What memory size is loaded will also be printed automatically. (This is a good idea if you run a Tasker Buss - you'll soon know if one of your boards has come loose again.)

I hope you will find this BEXEC\* as useful as I have.

```
2 REM      SUPER BEXEC*  FOR  COM-DOS 1.2
4 REM      UPDATE JAN 1983
6 REM      BY  ROB BRETTERECKER
8 REM      FOR NEW-DOS  OR  OLD-DOS
10 REM
12 X=PEEK(10950):POKE8993,X:POKE8994,X
14 PRINT"1) NEW DOS  2) OLD DOS":PRINT:PRINT:PRINT
16 INPUT"YOUR SELECTION";S:PRINT:IFS=2GOTO158
18 HM=PEEK(8960)
20 IF HM<159 THEN 36
22 DISK!"CA 9A00=15,1" : DISK!"GO 9A00
24 BS=40448
26 DISK!"CL":PRINT"1) 24*24  OR  2) 48*12":PRINT:PRINT:PRINT
28 PRINT"YOUR CHOICE":INPUTW$
30 IFW$="2"THENGOSUB134
32 DISK!"CL":PRINT"(40K VERSION LOADED)":PRINT
34 GOTO 70
36 IF HM<127 THEN 52
38 DISK!"CA 7A00=16,1" : DISK!"GO 7A00
40 BS=32256
42 DISK!"CL":PRINT"1) 24*24  OR  2) 48*12":PRINT:PRINT:PRINT
44 PRINT"YOUR CHOICE":INPUTW$
46 IFW$="2"THENGOSUB134
48 PRINT"(32K VERSION LOADED)":PRINT
50 GOTO 70
52 IF HM<95 THEN 68
54 DISK!"CA 5A00=17,1" : DISK!"GO 5A00
56 BS=24064
58 DISK!"CL":PRINT"1) 24*24  OR  2) 48*12":PRINT:PRINT:PRINT
60 PRINT"YOUR CHOICE":INPUTW$
62 IFW$="2"THENGOSUB134
64 PRINT"(24K VERSION LOADED)":PRINT
66 GOTO 70
68 PRINT"NO DOS AVAILABLE FOR YOUR MEMORY SIZE
70 HM=HM-6: POKE 133,HM : POKE 8960,HM
72 CLEAR
74 POKE 2073,173 :POKE 2893,55 :POKE 2894,8
```

```

76 DIMN$(36)
78 C$="CA 2E79=12,"
80 P=11897
82 DEFFNA(X)=10*INT(X/16)+X-16*INT(X/16)
84 POKE8955,213:POKE8956,252
86 DISK!C$+"1"
88 GOSUB 96
90 DISK!C$+"2"
92 GOSUB 96
94 GOTO122
96 FORI=PTOP+248STEP8
98 IFPEEK(I)=35THEN118
100 A$=STR$(N+1)
102 M$=MID$(A$,2)
104 N$=""
106 FORX=ITOI+5
108 N$=N$+CHR$(PEEK(X))
110 NEXTX
112 N$(N)=N$
114 N=N+1
116 PRINTM$TAB(3)N$TAB(12)FNA(PEEK(I+6))TAB(16)FNA(PEEK(I+7))
118 NEXTI
120 RETURN
122 PRINT:INPUT"YOUR SELECTION";X
124 IFX=0THENNEW
126 IFX=1THENPRINT:PRINT"YOU CAN'T RUN 'OS65D3'":GOTO122
128 IFX=4THENPRINT:PRINT"YOU CAN'T RUN 'DOS1.2'":GOTO122
130 IFX>VAL(M$)THEN PRINT:PRINT"THERE 'R ONLY 'M$' PROGRAMS":GOTO122
132 RUNN$(X-1)
134 DISK!"CL": POKE9730,16:POKE9743,15:POKE9723,63:POKE9736,63
136 POKE9725,10:POKE9738,59:POKE9800,64:POKE9636,75
138 POKE9766,75:POKE9770,75:POKE9815,75:POKE9670,123
140 POKE9783,123:POKE9682,164:POKE55296,1
142 FORI=1TO16:PRINT
144 IF VS=1 THEN RETURN
146 POKEBS+59,64:POKEBS+179,122:POKEBS+205,75
148 POKEBS+241,64:POKEBS+272,234:POKEBS+274,236
150 POKEBS+280,16:POKEBS+289,234:POKEBS+291,40
152 POKEBS+297,16:POKEBS+324,75
154 POKEBS+3,75
156 RETURN
158 POKE2073,173:POKE2893,55:POKE2894,8

```

---

#### FOR SALE

SUPERBOARD II, Series II, with 8K of RAM, DABUG III monitor, 300/600 baud cassette, 5Volt 10Amp professional power supply, B&W Philips T.V., cassette recorder, all manuals, in metal case with wooden sides, plus 100's of magazine articles, data, etc. on the computer and 100's of BASIC and machine code games, educational and business programs. The lot \$500.00  
Phone John Tee

3 Tasker memory boards - \$50.00 each. 1 Tasker Eprom board with Eproms programmed with Assembler and Extended monitor - \$25.00 1 Tasker Via/Pia board - \$35.00 1 model 15 teletype with transformers - \$60.00  
Phone John Tee

COMP-DOS 1.3  
*by Michael Lemaire & Paul Dodd*

An overview (or underview, or sideview or....)

COMP-DOS 1.3 is a revision of COMP-DOS 1.2, with many extra features, with all (?) of the bugs of V1.2 remedied. It has:-

- 14 character filenames
- improved directory format, including heading and two-column display
- "SET PROMPT" command for the DOS kernel. This will enable you to change the prompt from the rather mundane "A\*" to elegant messages of the calibre of "Yes, Sir?" or "-->" or "WOT?".
- implicit disk drive selection; eg. DR%B will give the directory of drive B without changing the currently Selected drive; eg. "CR GOLGAFRINCHAN,06%B"
- filename wildcards using the character "?" eg. "COP?" matches "COPIER" and "COPSLOP" eg. "LO GOL?" will load "GOLgafrinchan"
- proper error messages, eg. "ERROR #6 - drive not ready".

As COMP-DOS 1.3 works with OS65D version 3.3, two useful extras are available:-

- a decent keyboard driver with proper lowercase operation. All BASIC and DOS commands may be entered in lowercase.
- a real video driver with -
  - full colour RAM handling; the user can set output to a certain colour eg. inverse, and scrolling shifts the colour RAM
  - cursor X,Y location
  - clear screen, clear-to-end-of-screen, home commands
  - windowing
  - insert line, delete line, reverse scroll.
- A powerful line editor available for BASIC program editing
- BASIC "TRAP" command.

COMP-DOS 1.3 with OS65D V3.3 is available in the standard 24K/32K/40K version, and in a 48K version. We can supply any radical memory size you might have. The system will cost \$35.00 if you don't have a legal COMP-DOS 1.2, or \$15.00 if you do. (A couple of cases of Fosters might sway our decision.)

The source is not yet available because we don't want to give it to you, but the software can be bought from COMP-SOFT, 235 Swan St, Richmond, 3121, phone 03 428 5269

---

NOTES FROM FRANK HALLEY

Re Rodney Eisfelder's item on improving the maths functions in disk BASIC (Newsletter 2/8 page 13). There are two other changes that need to be made:- \$1845 should be changed to B0 1A and \$18F6 should be changed to 4C 54 19. Without these changes, some operations like "PRINT 2↑2" will cause a crash.

I am currently working on some code to put into the areas freed by Rodney's modifications to the maths operation. The code will significantly improve operations on random-access disk files, in the following ways:

1. DISK GET,n will now only read in a new track if necessary. If a read is necessary, a write of the old track in the buffer will be done only if necessary. This has the same effect as Nigel Bisset's article in newsletter 3/5 but is taken care of by the sytem. DISK PUT is now optional - it is still a valid command but now doesn't do anything.

*Next page please.*

HELP UTILITY for fig-FORTH  
*by Ray Gardiner*

This is a utility programming aid that will allow the dictionary structure to be displayed to locate compiling errors.

When you are compiling from disk, or testing new definitions nothing is more useful than a quick dictionary snapshot, this utility will tell you which word the compile error occurred in.

The code is based loosely on the W.F. Ragsdale 'HELP' listed in FORTH DIMENSIONS Vol.1,2, to operate, type 'HELP' after the error occurs, and hit the break key (escape key).

The output is formatted for the 64 screen but could be shortened for other screen formats.

Example of the output generated by 'HELP'

```
EDIT      HEADER 1BBC=84   LINK=1BA6   PFA 1BC5
CLR       HEADER 1BA6=83   LINK 1B93   PFA 1BAE
EDITOR    HEADER 1B93=C6   LINK 1B82   PFA 1B9E
CODE      HEADER 1B82=84   LINK 1B6C   PFA 1B8B
ASSEMBLER HEADER 1B6C=89   LINK 1B4B   PFA 1B7A
.....the listing continues to display.....
              the dictionary structure until you
              hit the escape key.
```

Included in the screen listing is a revised form of non-destructive stack print:

```
SCR# 6
(HELP UTILITY FOR fig-FORTH      )
1 FORTH DEFINITIONS HEX   06 USER SO
2 : DEPTH SO @ SP@ - 2 / 1 - ;
3 : .S CR DEPTH IF SP@ 2 - SO @ 2 - DO I @ . -2 +LOOP
4   ELSE ." EMPTY " THEN ;
5 0 VARIABLE #NFA 0 VARIABLE #PFA 0 VARIABLE #LFA
6 : @NEXT #LFA @ DUP #NFA ! PFA DUP #PFA ! DUP CFA #CFA !
7   LFA @ #LFA ! ;
8 : *HEAD ." HEADER " #NFA @ DUP 4 U.R ." =" C@ 2 U.R ;
9 : *PFA ." PFA " #PFA @ 4 U.R ;
10 : *LINK ." LINK " #LFA @ 4 U.R ;
11 : HELP BASE C@ >R LATEST #LFA ! CR BEGIN @NEXT #NFA @
12   DUP ID. COUNT OF AND 12 SWAP - SPACES DROP *HEAD *LINK
13 *PFA CR #LFA @ 0= ?TERMINAL OR UNTIL R> BASE C! ;
14 DECIMAL ;S                      (R. GARDINER DEC. 1982)
```

---

2. DISK CLOSE,n will now only write out the buffer contents if they have been altered.

3. A new command DISK RECORD,n will be added to assign the length of records in random access files after the file has been opened. 'n' must be from 1 to 2048. This replaces the two pokes necessary to alter record lengths from the standard 128 bytes.

I have written and tested the code, but at the moment it won't fit into the available free memory, and I'm working on compressing it. If anyone is interested in collaborating with me on the project you could write to me at